RZxS シリーズ用 Modbus Driver
（形式：RZOCX）
取扱説明書

（本取扱説明書は OCX アプリケーションのバージョン Ver. 1.0.0 に対応しています）

ＮＭ－５６５１－Ａ 改１

目　次

1.　はじめに

本書は、「RZxS シリーズ用 Modbus Driver : RZOCX」の取り扱い方法、操作手順、注意事項などを説明したものです。Windows 上の OLE コンテナアプリケーションである Visual Basic, Excel などの操作や用語を理解している方を前提にしています。アプリケーション Visual Basic, Excel などの操作や用語については、それぞれのマニュアルを参照してください。

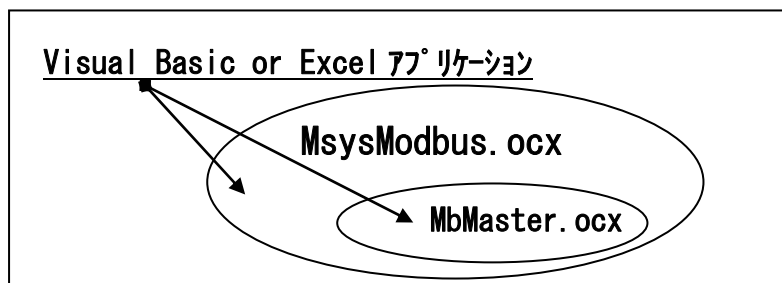
なお、本文中の MsysModbus.ocx、MbMaster.ocx を総称して RZOCX と呼んでいます。

1.1.　MsysModbus.ocx とは

一般に、ActiveX コントロール（OCX）とは、Windowsにおいてアプリケーションソフト間のデータ連携を実現する仕組みであるOLEに準拠して作成されたソフトウェア部品です。単独で実行することはできず、アプリケーションソフトに組み込んで使用します。

Msystem Modbus ActiveX Control（MsysModbus.ocx）は、Visual Basic や Excel などの OLE コンテナアプリケーション上で､PC の COM ポートに接続された Modbus デバイスのデータに迅速にかつ容易にアクセスすることを可能にします。 接続は、複数の Modbus デバイスにアクセスでき、非同期動作が可能です。

MsysModbus.ocx は、Modbus Serial 通信のデータ構造や通信の仕組み、通信のエラー処理、また、Modbus デバイスの各種データのアドレスを知る必要なく､Modbus Serial 通信の Open/Close やデータの Read/Write を容易に行なえるようにします。

また、MsysModbus.ocx は、アドレス指定で 1 byte データにアクセスできる MbMaster.ocx を含む形式で作成されています。それゆえ、MsysModbus.ocx を組み込んだアプリケーション上では、MbMaster.ocx のコントロールも直接利用することができます。MbMaster.ocx のコントロールに関しては、Modbus Master ActiveX Control 取扱説明書に記述してありますので、ご覧下さい。

1.2.　MsysModbus.ocx のレジスタ登録

MsysModbus.ocx のコントロールをアプリケーションで使う前に、Windows レジストリーに MsysModbus.ocx と MbMaster.ocx を登録する必要があります。以下の例のように、DOS コマンドで RegSvr32.exe の引数にフルパスのコントロール名を与えることで､登録することができます｡ただし、各 ocx と一緒にダウンロードされる modbusm.dll ファイルが、ocx ファイルと同一フォルダ内に必要です。

```
Regsvr32 C:¥xxxxx¥xxxxx¥MsysModbus.ocx
Regsvr32 C:¥xxxxx¥xxxxx¥MbMaster.ocx
```

簡単な登録方法として、SampleProj フォルダ内にバッチファイル MsysReg.bat が用意してあります。MsysModbus.ocx と MbMaster.ocx の両方のレジスタ登録を一緒に行うことができます。
レジスタ登録“succeeded”を示すメッセージがそれぞれ出力されますので､OK をクリックして下さい。

1.3.　動作環境

MsysModbus.ocx をお使いいただくためには，以下のハードウェアとソフトウェアが必要です。

- Windows2000,XP が正しくインストールされた DOS/V 互換パーソナルコンピュータ。
- Visual Basic や Excel などの Windows 上で動作するソフトウェア

1.4.　免責事項

サンプルソースファイルのコーディング、その他の内容に関するご質問等のお問い合わせには対応いたしませんので、ご了承ください。また、本ソフトウェアは無償で提供されますが、本製品の保守、保証は行わないものとします。

2. Msystem Modbus コントロール説明

2.1. 共通コントロール

各種 Modbus デバイスへ共通にアクセスできるコントロールです。

2.1.1. Modbus_Serial_Connect()

Modbus_Serial_Connect(MyHandle As Long, CommPort As Integer,
BaudRate As Integer, Parity As Inetger) As Integer

戻り値　　通信接続時に得たステータスコード（別表参照）を返します。

引数　　MyHandle :　　接続時のハンドル番号を受け取ります。
CommPort :　　Comm ポート番号（1,2,3,..）を指定します。
BaudRate :　　ボーレート（9600,19200,38400）を指定します。
Parity :　　パリティー（0:None、1:Odd、2:Even）を指定します。

解説　　Modbus Serial 通信を接続します。

```
例　　Dim　MyHandle As Long
　　　MsysModbus1.Modbus_Serial_Connect(MyHandle, 1, 38400, 1)
```

2.1.2. Modbus_Serial_Disconnect()

Modbus_Serial_Disconnect(MyHandle As Long) As Integer

戻り値　　通信切断時に得たステータスコードを返します。

引数　　MyHandle :　　切断する通信のハンドル番号を指定します。

解説　　Modbus Serial 通信を切断する

```
例　　MsysModbus1.Modbus_Serial_Disconnect(MyHandle, 1, 38400, 1)
```

2.2.　RZUS 用コントロール

RZUS、RZMS（以下 RZUS と記述）デバイスにアクセスできるコントロールです。

2.2.1.　RZUS_Info_Read()

RZUS_Info_Read(MyHandle As Long, Slave As Integer, InfoDat() As String)
As Integer

戻り値　　データ読み出し時に得たステータスコードを返します。

引数　　MyHandle：　通信のハンドル番号を指定します。
Slave：　スレーブアドレスを指定します。
InputDat：　製品情報を受け取ります。
InfoDat(0) = 形式（Model Name）
InfoDat(1) = 機番（Serial No.）
InfoDat(2) = ﾊｰﾄﾞｳｪｱﾊﾞｰｼﾞｮﾝ番号（Hard Version）
InfoDat(3) = ｿﾌﾄｳｪｱﾊﾞｰｼﾞｮﾝ番号（Soft Version）
InfoDat(4) = 製造日（Manufact. Date）
InfoDat(5) = 校正日（Calib. Date）

解説　　接続しているデバイスの製品情報を読み出します。

```
例　Dim　InfoDat() As String
　　MsysModbus1.RZUS_Info_Read(MyHandle, 1, InfoDat)
```

2.2.2.　RZUS_Type_Read()

RZUS_Type_Read(MyHandle As Long, Slave As Integer, TypeDat() As Integer)
As Integer

戻り値　　データ読み出し時に得たステータスコードを返します。

引数　　MyHandle：　通信のハンドル番号を指定します。
Slave：　スレーブアドレスを指定します。
TypeDat：　入力タイプ情報（ﾁｬﾈﾙ 1～ﾁｬﾈﾙ 12）を受け取ります。
TypeDat(i) = ﾁｬﾈﾙ i+1 の入力ﾀｲﾌﾟ（Input Type）
入力ﾀｲﾌﾟｺｰﾄﾞは 10 進数

解説　　入力タイプ情報（ﾁｬﾈﾙ 1～ﾁｬﾈﾙ 12）情報を読み出します。

```
例　Dim　TypeDat() As Integer
　　MsysModbus1.RZUS_Type_Read(MyHandle, 1, TypeDat)
```

2.2.3.　RZUS_AI_Read()

RZUS_AI_Read(MyHandle As Long, Slave As Integer, AIDat() As Single)
As Integer

戻り値　　データ読み出し時に得たステータスコードを返します。

引数　　MyHandle：　通信のハンドル番号を指定します。
Slave：　スレーブアドレスを指定します。
AIDat：　基本入力データ（ﾁｬﾈﾙ 1～ﾁｬﾈﾙ 12）を受け取ります。
AIDat(i) = ﾁｬﾈﾙ i+1 の基本入力ﾃﾞｰﾀ（Real Input）

解説　　基本入力データ（ﾁｬﾈﾙ 1～ﾁｬﾈﾙ 12）情報を読み出します。

```
例　Dim　AIDat() As Single
　　MsysModbus1.RZUS_AI_Read(MyHandle, 1, AIDat)
```

2.2.4. RZUS_AIStatus_Read()

RZUS_AIStatus_Read(MyHandle As Long, Slave As Integer,
AIStatusDat() As Integer) As Integer

戻り値　データ読み出し時に得たステータスコードを返します。

引数　MyHandle：　通信のハンドル番号を指定します。
Slave：　スレーブアドレスを指定します。
AIStatusDat：　基本入力ｽﾃｰﾀｽ（ﾁｬﾈﾙ 1～ﾁｬﾈﾙ 12）を受け取ります。
AIStatus(i) = ﾁｬﾈﾙ i+1 の基本入力ｽﾃｰﾀｽ（AI Status）

解説　基本入力ｽﾃｰﾀｽ（ﾁｬﾈﾙ 1～ﾁｬﾈﾙ 12）情報を読み出します。

例
```
Dim　AIStatusDat() As Integer
MsysModbus1.RZUS_AIStatus_Read(MyHandle, 1, AIStatusDat)
```

2.2.5. RZUS_DI_Read()

RZUS_DI_Read(MyHandle As Long, Slave As Integer, DIDat As Integer)
As Integer

戻り値　データ読み出し時に得たステータスコードを返します。

引数　MyHandle：　通信のハンドル番号を指定します。
Slave：　スレーブアドレスを指定します。
DIDat：　ﾄﾘｶﾞ接点入力（Trigger Input）を受け取ります。
値 0 は Off／値 1 は On

解説　ﾄﾘｶﾞ接点入力情報を読み出します。

例
```
Dim　DIDat As Integer
MsysModbus1.RZUS_DI_Read(MyHandle, 1, DIDat)
```

2.2.6. RZUS_DO_Read()

RZUS_DO_Read(MyHandle As Long, Slave As Integer, DODat As Integer)
As Integer

戻り値　データ読み出し時に得たステータスコードを返します。

引数　MyHandle：　通信のハンドル番号を指定します。
Slave：　スレーブアドレスを指定します。
DODat：　警報接点出力（Alarm Output）を受け取ります。
値 0 は Off／値 1 は On

解説　警報接点出力情報を読み出します。

例
```
Dim　DODat As Integer
MsysModbus1.RZUS_DO_Read(MyHandle, 1, DODat)
```

2.2.7. RZUS_Type_Write()

RZUS_Type_Write(MyHandle As Long, Slave As Integer, TypeDat() As Integer)
As Integer

戻り値　　データ書き込み時に得たステータスコードを返します。

引数　　MyHandle：　通信のハンドル番号を指定します。
Slave：　スレーブアドレスを指定します。
TypeDat：　入力タイプ情報（ﾁｬﾈﾙ 1～ﾁｬﾈﾙ 12）を指定します。
TypeDat(i) = ﾁｬﾈﾙ i+1 の入力ﾀｲﾌﾟ（Input Type）
入力ﾀｲﾌﾟｺｰﾄﾞは 10 進数

解説　　入力タイプ情報（ﾁｬﾈﾙ 1～ﾁｬﾈﾙ 12）情報を書き込みます。

```
例　Dim　TypeDat() As Integer
    For i = 1 To 12
        TypeDat(i) = 5
    Next i
    MsysModbus1.RZUS_Type_Write(MyHandle, 1, TypeDat)
```

2.2.8. RZUS_DO_Write()

RZUS_DO_Write(MyHandle As Long, Slave As Integer, DODat As Integer)
As Integer

戻り値　　データ書き込み時に得たステータスコードを返します。

引数　　MyHandle：　通信のハンドル番号を指定します。
Slave：　スレーブアドレスを指定します。
DODat：　警報接点出力（Alarm Output）を指定します。
値 0 は Off／値 1 は On

解説　　警報接点出力情報を書き込みます。

```
例　MsysModbus1.RZUS_DO_Write(MyHandle, 1, 1)
```

2.3.　ステータスコード一覧表

エラーコード	内容
0	正常
N < 255	Slave Device Exception Response N = Exception Value
256	Invalid Connection Handle
257	Message Overrun
258	Invalid Point Address
259	Invalid Slave Node Address
260	Invalid Length
261	Unsupported Modbus Command
263	Slave Device Time-Out
264	Invalid Transmission Mode
265	Invalid CRC In Slave Response
266	Connection Not Established
267	Invalid Slave Response
271	Demo Time Expired
272	Invalid modbus/TCP Command

3. パフォーマンスについて

3.1. 各コントロールのパフォーマンス

OS : Windows2000、CPU : Pentium(R)4　1.60GHz にて測定した結果を参考用として下記に示します。各パフォーマンスは、動作するパソコンの OS、環境、ドライバ等に大きく依存します。実際の使用に関しては、必ず、動作用パソコンにて実測値をご確認ください。

Modbus_Serial_Connect()	: 39 msec.
Modbus_Serial_Disconnect()	: 361 msec.
RZUS_Info_Read()	: 50 msec.
RZUS_Type_Read()	: 20 msec.
RZUS_AI_Read()	: 31 msec.
RZUS_AIStatus_Read()	: 15 msec.
RZUS_DI_Read()	: 11 msec.
RZUS_DO_Read()	: 261 msec.
RZUS_Type_Write()	: 213 msec.
RZUS_DO_Write()	: 254 msec.

3.2. CPU 負荷率

MsysModbus コントロールは、通信パフォーマンスをあげるため、各コントロールの中には CPU 負荷率を開放する仕組みをは入れていません。それゆえ、周期的にコントロールを呼び出すプログラミングをしたアプリケーションは、CPU 負荷率が高くなってしまいます。

Visual Basic のサンプルプログラムのように、コントロールの呼び出しと呼び出しの間に、適宜に Sleep()関数を使用すると、CPU を他アプリケーションに開放することができ、CPU 負荷率を下げることが可能です。必要に応じてご対処ください。